6

UNITED STATES DEPARTMENT OF COMMERCE
Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

MF

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.
09/030,258	02/25/98	SCHULTZ	R 12217-100

027267 TM02/0619
WIGGIN & DANA
ATTENTION: PATENT DOCKETING
P.O. BOX 1832
NEW HAVEN CT 06508-1832

EXAMINER	
LAO, S	
ART UNIT	PAPER NUMBER
2151	17

DATE MAILED: 06/19/01

Please find below and/or attached an Office communication concerning this application or proceeding.

Commissioner of Patents and Trademarks

1- File Copy

SM

Office Action Summary

Application No. 09/030,258	Applicant(s) Schultz, et al
Examiner S. Lao	Art Unit 2151

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136 (a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) ☒ Responsive to communication(s) filed on *Apr 9, 2001*

2a) ☒ This action is **FINAL.** 2b) □ This action is non-final.

3) □ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*1935 C.D. 11; 453 O.G. 213.

Disposition of Claims

4) ☒ Claim(s) *1-46 and 60-63* is/are pending in the applica

4a) Of the above, claim(s) ______ is/are withdrawn from considera

5) □ Claim(s) ______ is/are allowed.

6) ☒ Claim(s) *1-46 and 60-63* is/are rejected.

7) □ Claim(s) ______ is/are objected to.

8) □ Claims ______ are subject to restriction and/or election requirem

Application Papers

9) □ The specification is objected to by the Examiner.

10) □ The drawing(s) filed on ______ is/are objected to by the Examiner.

11) □ The proposed drawing correction filed on ______ is: a) □ approved b) □ disapproved.

12) □ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. § 119

13) □ Acknowledgement is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d).

a) □ All b) □ Some* c) □ None of:

1. □ Certified copies of the priority documents have been received.
2. □ Certified copies of the priority documents have been received in Application No. ______ .
3. □ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

*See the attached detailed Office action for a list of the certified copies not received.

14) □ Acknowledgement is made of a claim for domestic priority under 35 U.S.C. § 119(e).

Attachment(s)

15) ☒ Notice of References Cited (PTO-892)

16) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

17) □ Information Disclosure Statement(s) (PTO-1449) Paper No(s). ______

18) □ Interview Summary (PTO-413) Paper No(s). ______

19) □ Notice of Informal Patent Application (PTO-152)

20) □ Other:

DETAILED ACTION

1. Claims 1-46 and 60-63 are pending. This action is in response to the amendment filed 4/9/2001. Applicant has amended claims 1-6, 8-20, 24, 27-34, 42 and 45, and added claims 60-63.

2. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

3. Claims 1-4, 20-27, 34-35, 41-44, 46, 63 are rejected under 35 U.S.C. 103(a) as being unpatentable over Krishnamurthy et al in view of Mahajan.

As to claim 1, Krishnamurthy teaches data processing system (Yeast, event-based cooperative process management system), including

event modules (user, or tools or actions of Yeast specification, page 141, section 2.2.2) each including code that generates an event data signal (event announcements) representative of a particular event (announce events, page 134, section 2; page 137, section 2.2.2),

scripts (action) each having instructions (sequence of command, section 2, first para., section 3.2) that provides results (event announcements made by actions of Yeast),

processing modules (client, server) each including code that provides processed data (event) to said scripts (trigger action) (sections 2.2.2, 2.3); and

task module (command interpreter), selectively communicating with each of said event modules (client), including code for execution of a selected one of said scripts that corresponds to said event data signal (command interpreter to execute the action component triggered by event, page 134, section 2; section 2.3).

While Krishnamurthy modifies the course of actions, ie, process flow, by directly or indirectly incorporating results of previous event or action (event announcements made by user, or tools or actions of Yeast specification, page 141, section 2.2.2), Krishnamurthy does not explicitly teach (1) the task module selectively invokes processing modules according to status and results, (2) the modification includes incorporating the processed data to modify execution of next instructions of the selected script.

As to (1)-(2), Mahajan teaches a scripting system, wherein a task module (script interpreter 19) selectively invokes processing modules (exported functions referred to by scripts) according to status (<ConditionalStatements>) and results of previous execution (nested scripts, interpret scripts within another script), and incorporating the processed data (status = busy or ok, which results from MakeCall script execution) to modify execution of next instructions of the selected script (cause a script to be interpreted within another script). See col. 4, line 35 - col. 8, line 11. It is noted that Mahajan teaches two ways of modifying execution of next instructions of a script: CallScript and SetEvent. CallScript directly causes another script to be interpreted within a calling script, and SetEvent causes an event to be set which indirectly causes another script to be interpreted/executed. Both approaches modify execution of next instructions of a script by dynamically/conditionally invoking appropriate instructions represented by the another script based on previous execution results/data and status/conditions. See col. 6, line 59 - col. 7, line 4; col. 7, lines 33-52; col. 8, lines 5-11; col. 9, lines 13-20.

It would have been obvious to use the teachings of (1)-(2) of Mahajan in Krishnamurthy. This is because that Krishnamurthy interacts with users (page 137, section 2.2.2), which desires a user interface, and Mahajan provides a user interface. Therefore, one of ordinary skill in the art would have been motivated to combine Mahajan with Krishnamurthy.

As to claim 2, Krishnamurthy teaches implementing event-action based process management (fig. 1) in a high speed execution environment (Sun/Unix system, page 135), wherein the time difference between actions would have been very small, as such, the execution of scripts/actions would have been substantially simultaneous.

As to claim 3, Krishnamurthy teaches converter module to maps said event data signal to scripts upon reception (command interpreter, match the event component, page 134, section 2.3).

As to claim 4, Krishnamurthy teaches processing modules / task module as client / server.

As to claims 21-26, inherently, Krishnamurthy's system includes storage / computer-readable medium / persistent memory for storing code. Since the system of Krishnamurthy interacts with user (page 137, section 2.2.2), including a standard language interface or a graphical user interface would have been inherent. Script building module for creating scripts is met by Krishnamurthy (generating specification, page 141, section 3.2, first para.).

As to claims 20, 27, Krishnamurthy teaches (page 137, section 2.2.2, page 133, second para.) actions are inter-related by events, and events trigger actions which in tern trigger further event announcement. As such, scripts/actions are modified dynamically / at run time. Further, Mahajan teaches scripts is preprogrammed to iteratively/dynamically update/modify its contents (see discussion of claim 1 with respect to Mahajan).

As to claim 34, it is basically a method claim of claim 1. Note claim 1 for rejection.

As to claim 35, note discussion of claims 13-19 and 32 for communication interface and peripherals.

As to claims 41-44, note discussion of claims 13-19 for protocols and interfaces and claim 2 for substantially simultaneously. Accessing only the peripheral modules that capable of performing processing operations is a typical part of load balancing.

As to claim 46, providing results of execution is taught by Waclawsky (monitor performance).

As to claim 63, it is covered by claim 1 except for first event and second event, which is taught by Mahajan in that a first event is met by the event which triggers a (calling) script and a second event is met by event produced by SetEvent function. Col. 9, lines 13-20. Note discussion of claim 1 for motivation to combine.

4. Claims 5-19, 28-33, 36-40, 45, 60-62 are rejected under 35 U.S.C. 103(a) as being unpatentable over Krishnamurthy et al in view of Mahajan as applied to claim 1 and further in view of Waclawsky et al.

As to claims 5-6, 8-9, Waclawsky teaches event-based network management (event driven interface, abstract), including providing information relating to operating conditions (performance measure, step 408) and load balancing (load balancing, modify network operation) (abstract, step 412). Direct communication is taught by the network configuration.

It would have been obvious to use the network management features of Waclawsky in Krishnamurthy. This is because that the system of Krishnamurthy is a network system (client-server system), which in operation requires a network management system. Waclawsky provides a network management system. Therefore, one of ordinary skill in the art would have been motivated to use the network management system as taught by Waclawsky in the system of Krishnamurthy to manage the network operations.

As to claim 7, storing script/specification is taught by Krishnamurthy as modified (Mahajan, 20-22, fig. 3).

As to claims 10-12, Krishnamurthy as modified teaches (Waclawsky) bidirectionally and substantially simultaneously transmitting data between (network), dynamically assigning processing functions (compare performance and modify network operation, steps 408, 410, 412).

As to claims 13-19, Krishnamurthy as modified teaches (Waclawsky) communication interfaces (event driven interface) and protocols (method/system of Waclawsky) between various modules of the network.

As to claims 28-32, Krishnamurthy as modified teaches (Waclawsky) protocols and communication interfaces (note discussion of claims 13-19 above), means for transmitting and receiving response data (client/server), and peripherals (printer 26).

As to claim 33, note discussion of claim 1 and Krishnamurthy as modified teaches (Waclawsky) resource management module that dynamically assigns processing functions to (media manager 102); and administrative module that receives and presents data relating to (network monitor 22). Fig.s 1, 6, 10.

As to claim 36, note discussion of claim 11.

As to claims 37-40, Waclawsky teaches producing response data signals as a result of execution (monitor performance); transmitting response data signals from a task module to selected said peripheral modules (output control signal, step 412), storage (memory 100). As to the step of translating, data formatting/translating is common practice in the art when a sender and a receive have different formats/conventions.

As to claim 45, note discussion of claims 20 and 27.

As to claim 60, it is covered by claim 1 except for response profile which is met by Waclawsky (trace, col. 1, lines 43-67). A program trace is typically built by recording events during a program/process' execution and passed to the program after execution for debugging purposes. A well known example may be found in core dump in Unix or transaction log of Tuxedo.

As to claim 61, a program trace typically contains program/process events.

As to claim 62, Waclawsky teaches tracing execution (trace, col. 1, lines 43-67). Continuing execution from a last traced instruction after failure is taught by the well known roll-back protocol of transactional processing.

5. Applicant's arguments filed 4/9/2001 have been fully considered but they are not persuasive.

Applicant argued (page 13-15) that Krishnamurthy and Mahajan do not teach scripts that *dynamically invoke processing* in accordance with status and process data generated by previously executed instructions and modify execution of next instructions by incorporating data processed by the dynamically invoked processing modules (page 15), whereas applicant's invention provides *dynamic execution* of instructions and processing modules to gather data in response to event data, as disclosed in the specification as filed, on page 13, lines 3-9.

The examiner's response is as follows. Firstly, *dynamically invoke processing* in accordance with status and process data generated is not claimed. In stead, independent claims 1, 33, 60 and 63 requires *selectively invoking* (or selecting and invoking) in accordance with status and process data / results generated, whereas independent claim 34 only requires *invoking* in accordance with status.

Secondly, *dynamically invoke processing* does not appear to be disclosed in the application as filed. The support cited by applicant (page 13, lines 3-9) is as follows:

"The script may have a memory that *dynamically keeps track* of modifications to the script itself. As the steps are being performed, affected application programs receive accurate and *dynamic information* regarding the status of other processing modules and result data generated. The script incorporates the information into the instructions and thereby modifies the action taken by the script." (emphasis added)

Clearly, as disclosed, the dynamic nature concerns record keeping such that the status information and/or result data are dynamic/updated information. Dynamically invocation is not discussed in this passage cited by applicant. Further, as disclosed, selectively invoking, or modifying, refers to selecting a script to execute based on event/input data. Once the script is chosen, it is retrieved and executed as written/coded. See application, page 35-36, description of fig. 3F. In other words, it is the selection of next script that is based on event input that is updated, ie, dynamic. The coding of script chosen is not changed. In terms of process control, it is the flow of process steps that is dynamic, but the process steps themselves (as defined by corresponding scripts) remains the same.

Thirdly, selectively invoking (or selecting and invoking) in accordance with status and process data / results generated as claimed is met by the combination of Krishnamurthy and Mahajan. As discussed on claim 1. In particular, Mahajan teaches a scripting system, wherein a task module (script interpreter 19) selectively invokes processing modules (exported functions referred to by scripts) according to status (<ConditionalStatements>) and results of previous execution (nested scripts, interpret scripts within another script), and incorporating the processed data (status = busy or ok, which results from MakeCall script execution) to modify execution of next instructions of the selected script (cause a script to be interpreted within another script). See col. 4, line 35 - col. 8, line 11. It is noted that Mahajan teaches two ways of modifying execution of next instructions of a script: CallScript and SetEvent. CallScript directly causes another script to be interpreted within a calling script, and SetEvent causes an event to be set which indirectly causes another script to be interpreted/executed. Both approaches modify execution of next instructions of a script by dynamically/conditionally invoking appropriate instructions represented by the another script based on previous execution results/data and status/conditions. See col. 6, line 59 - col. 7, line 4; col. 7, lines 33-52; col. 8, lines 5-11; col. 9, lines 13-20.

Applicant argued that Waclawsky is not properly combined because it is non-analogous art and the combination is based on hindsight. (Page 16-17).

Regarding the argument of non-analogous art, the examiner's position is that Waclawsky is analogous art to Krishnamurthy and Mahajan because they all directed to process flow control in general and event processing in particular.

In response to Applicant's argument that the Examiner's conclusion of obviousness is based upon improper hindsight reasoning, it must be recognized that any judgement on obviousness is in a sense necessarily a reconstruction based upon hindsight reasoning. But so long as it takes into account only knowledge which was within the level of ordinary skill at the time the claimed invention was made, and does not include knowledge gleaned only from the applicant's disclosure, such a reconstruction is proper. In re McLaughlin, 443 F.2d 1392; 170 USPQ 209 (CCPA 1971).

For these reasons above, applicant's arguments are not persuasive.

6. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for response to this final action is set to expire THREE MONTHS from the date of this action. In the event a first response is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action.

In no event will the statutory period for response expire later than SIX MONTHS from the date of this final action.

7. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Sue Lao whose telephone number is (703) 305-9657. A voice mail service is also available at this number. The fax phone number for the organization where this application or proceeding is assigned is (703) 308-9051 for regular communications and After Final communications.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the Group receptionist whose telephone number is (703) 305-9600.

Sue Lao
June 14, 2001

MAJID A. BANANKHAH
PRIMARY EXAMINER